Unclassified

PROGRESS ON A BASIC ANALYSIS OF THE ERRORS IN ANALOGUE COMPUTERS

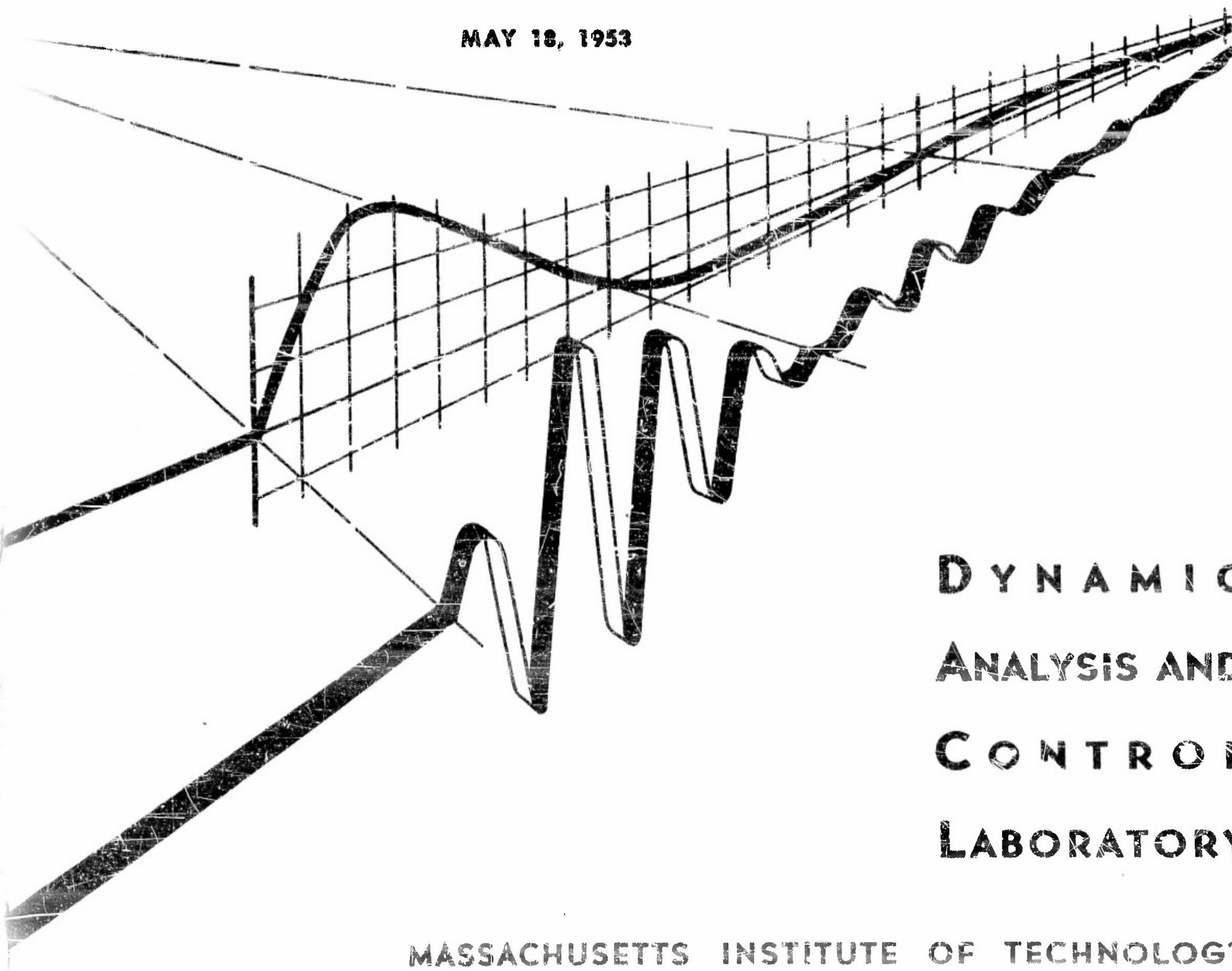
by

Max V. Mathews

D. A. C. L. RESEARCH MEMORANDUM NO. R. M. 7023-1

COPY NO. 1

MAY 18, 1953



DYNAMIC ANALYSIS AND CONTROL LABORATORY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE • 39 • MASSACHUSETTS

Unclassified

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

DYNAMIC ANALYSIS AND CONTROL LABORATORY

D. A. C. L. RESEARCH MEMORANDUM NO. R. M. 7023-1

PROGRESS ON A BASIC ANALYSIS OF THE ERRORS IN ANALOGUE COMPUTERS

by

MAX V. MATHEWS

PREPARED FOR

THE OFFICE OF NAVAL RESEARCH

MAY 18, 1953

CAMBRIDGE 39, MASS.

TABLE OF CONTENTS

FOREWORD

The increase in recent years of the importance of analogue computation as an effective tool in engineering analysis has been accomplished by a growth in the complexity of the problems being solved on computers. Because this complexity has made more difficult and expensive the operation of computers, knowledge of the errors in computer solutions becomes necessary. Unfortunately, the increased problem complexity makes much more difficult the estimation of the accuracy of solutions.

In the operation of the M.I.T. Flight Simulator, the staff of the Dynamic Analysis and Control Laboratory at the Massachusetts Institute of Technology has recognized that the future contributions made by analogue-computing facilities depend to a large extent upon whether computational errors can be predicted, measured, and evaluated. Discussions between the staff of the D.A.C.L. and Prof. F. J. Murray of Columbia University led to a proposal from the D.A.C.L. to the Office of Naval Research. This proposal, which resulted in contract N5ori-07879, outlined a study with the objectives of "(1) the development of methods for determining the feasibility of solving a given problem to a specified degree of accuracy on a particular machine and (2) the development of methods for determining the accuracy of solutions while they are being studied on a particular computer." The present report covers research on this program from May 16, 1952 to May 15, 1953. Much of this research has been accomplished as S.M. theses at M.I.T., and appropriate references indicate the staff members whose contributions constitute the research.

Prepared by:	Approved by:
M. V. Mathews	W. W. Seifert
M. V. Mathews Research Assistant	W. W. Seifert Assistant Director

ABSTRACT

The work accomplished during the past year at the Dynamic Analysis and Control Laboratory on error analysis for analogue computers is summarized and illustrated with examples. Various methods of error analysis are examined to determine how they combine to form an effective error-analysis tool. Methods are presented for testing computer components and for statistically describing their errors. Consideration has been given to components that are essentially linear feedback systems but that include nonlinearities such as backlash and limiting. The propagation of errors in typical computer problems is examined through the solution of a set of linear differential equations which approximately specify the error propagation. The linearizing approximations are justified experimentally for the problems considered.

The error propagation in a computer is examined experimentally and analytically. Calculated errors, determined by linearization approximations, are compared with observed errors. Practical checking methods to assure the proper operation of a computer and to locate faulty components and setup mistakes are described. In conclusion, suggestions for future work are outlined.

1. INTRODUCTION

The development of practical methods for predicting and evaluating computer errors is of considerable importance because the value of the solutions obtained from an analogue computer depends to a large extent on how well computational errors can be determined. The group operating the M.I.T. Flight Simulator at the Dynamic Analysis and Control Laboratory is particularly interested in this problem of error analysis. Consequently, members of this group have been engaged over the past two years in a program designed to lead to a better understanding of the nature of error generation and propagation in analogue computers. A full statement of the error-analysis problem as originally envisioned by the Dynamic Analysis and Control Laboratory is given in a proposal which was submitted to the Office of Naval Research in March, 1952. The proposal is summarized in Sec. 1.1.

Most of the work included in this report was accomplished as thesis research at the D.A.C.L. In particular the work of M. Mathews,[1*] H. Mori,[2] G. Rabow,[3] and N. Trembath[4] will be summarized. Although some significant results and a much better understanding of the over-all problem have been obtained during the first year of work under this program, progress on the task as originally outlined has not been as great as anticipated for two principal reasons. First, an unexpected shortage of personnel has limited the number of man-hours which could be allocated to the program and thus considerably restricted the rate of progress. Second, several of the investigations produced negative results, showing that some of the objectives as outlined in the original proposal were impractical. While the negative results are worth obtaining, their occurrence has made necessary the reorientation portions of the basic program. At present this reorientation has not been completed because some basic problems must be resolved before the direction for future research can be decided.

1.1. Brief Restatement of the Problem.

The original proposal outlined certain specific areas which would be investigated under the error-analysis program. Though they are not divided precisely in this way in the original program, the work of the past year can be divided best as outlined in Sec. 1.11.

1.11. Areas of Investigation.

Component Errors.

In this work, methods were to be developed for testing and evaluating practical computer components in order to determine the errors generated by the components. In general, the errors were to be determined both as random functions and as functions of the specific input to the components. The errors were to be evaluated in a form which would

*References are listed in the Bibliography of Appendix A.

be useful as input data in the problem of determining the error propagation in a computer.

Problem Errors.

A second subject for investigation was the way in which errors propagate in a particular problem consisting of a set of differential equations. A satisfactory method of evaluating error propagation before a problem was set up on a computer would provide a way of seeing whether the computer could produce an acceptably accurate solution to the problem. Consequently, this study was to be divorced from the characteristics of any specific computer, if possible, so as to be generally applicable. Since a number of theories for the propagation of errors have already been developed,[5, 6] the major work in this area would consist in determining whether the theories actually could be applied to practical problems, and in simplifying the computations involved so that useful information could be obtained at a reasonable cost.

Computer Errors.

Here an experimental study of error propagation in an actual computing machine working on a specific problem was to be made. The propagation of both inherent errors generated by the computer components of the machine and artificial errors purposely injected into the machine was to be examined. By means of this study, methods were to be developed to evaluate better the errors in the machine solutions and to increase confidence in the accuracy of the machine results over a wider range of variation in problem parameters.

1.12. Additional Areas for Investigation.

Besides these studies, two problems which were not considered in the original proposal have been added to the error-analysis program. The first of these is an analysis of the Flight Table section of the M.I.T. Flight Simulator, and the second is a study of some practical operational techniques for improving computer efficiency.

Flight Table Report.

Prior to the inception of the error-analysis program, a report analyzing and evaluating the Flight Table section of the M.I.T. Flight Simulator had been approximately half completed. The report was not only to describe the Flight Table in specific detail but also was to serve as a general explanation of the design, construction, and evaluation of a complex computing component. The publication of the report was believed by the Dynamic Analysis and Control Laboratory and the Office of Naval Research to be an important contribution to the field of analogue computation. In consequence, part of the work of producing the report was added to the error-analysis program.

Practical Operational Techniques.

While the problems in the original error-analysis proposal are by no means [illegible] theoretical, they are not considered from a viewpoint as practical as that taken by a problem engineer faced with the operation of a computer on a specific problem. A review of operating experience at the D.A.C.L. showed that the efficiency of computer operation varies significantly as a function of the skill of the operator. In particular, an important part of the operating time is spent in finding and eliminating setup mistakes and faulty components, and the extent of this check-up time depends largely on the operator's skill. As a result, it was though worth while to study the trouble-shooting techniques used by good operators, and to analyze and codify some of the most successful techniques.

2. GENERAL OUTLINE OF PROGRESS

Before the details are given of the work on these studies, this section is presented to summarize over-all progress and to show how the studies do and do not combine to form a unified error-analysis tool.

2.1. Component Errors.

The study of errors generated by components has produced powerful and easily applied techniques for determining the average error produced by a component, given the statistical properties of the input to the component and the statistical properties of any noise generated inside the component itself. The methods are applicable to the majority of computer components that can be represented as essentially linear devices that are subject, however, to a number of inescapable nonlinear limitations such as saturation and stiction. The theories developed offer a means of evaluating the average errors in a component output and thus provide reasonable methods for evaluating the quality of a computer component. Nevertheless, the methods for calculating the instantaneous time function* of the output error resulting from a specified input to the component are not as yet practical.

Although now it is feasible to measure both the linear and nonlinear characteristics of most components well enough so that their errors with a specific input can be calculated by numerical-integration methods, such calculations take much too long to be of any practical value in error-analysis work. Most of the techniques for the analytic investigation of the propagation of errors require as data the errors expressed as instantaneous time functions. A practical way of calculating these time-function errors is one of the essential links still missing in the general process for evaluating computational errors. Nor does it seem that this difficulty will be overcome easily because the problem has

*An implied assumption here is that the independent variable of computation is time. The assumption is true for the majority of analogue computers.

been reduced essentially to one of calculation, and the calculations have been proven to be inordinately long, even with the help of the most modern computing techniques. This problem is one of the impasses delaying the error-analysis program.

2.2. Problem Errors.

Analytic investigations of the propagation of errors in a system of differential equations have shown that the approximations involved in the theory of these calculations are justified, if not in general, then at least for the systems of differential equations that were tried. The principal approximation involved is that the errors propagate according to a set of linear differential equations even though the problem consists of a set of nonlinear differential equations. Additional approximations are required for the solution of the linear differential equations.

The propagation of errors was examined in a hand-calculated numerical check which was computed for a problem carried out on the M.I.T. Flight Simulator. The error-propagation analysis gave the approximate shape and magnitude of the errors in the solution as functions of time. Since for this particular solution the actual errors were known, the agreement between the actual errors and the calculated errors could be measured. The agreement was as good as could be required for error-analysis work, thus justifying the approximations.

The problem studied was quite a bit simpler than problems now being set up on the M.I.T. Flight Simulator, but probably it is above the average complexity of those set up on the usual electronic differential-analyzer installation. Despite the fairly simple nature of the problem, calculating the errors took approximately 70 man-hours of work. While this amount of work might be shortened somewhat through the use of ingenious methods, one is forced to conclude that the calculation of errors seldom would be justified in dealing with practical problems. This conclusion together with demonstration of the theoretical feasibility of calculating errors is the principal result from this part of of the error-analysis work.

2.3. Computer Errors.

In the experimental investigation of the propagation of errors in an actual computer, roughly the same conclusions were reached as in the previous analytic investigation of problem errors; that is, while the theories for determining these errors are unquestionably sound, the practical calculations take so much time on the computer that only very rarely would the knowledge of an error be valuable enough to justify the cost of its determination. The complete evaluation of the errors for one machine solution took roughly 80 hours of machine time. The machine time of the M.I.T. Flight Simulator costs approximately $100 per hour; hence, the total cost of measuring the error is large. Also, such a time-consuming error measurement defeats the purpose of an analogue computer for most types of problems, where its great advantage is its ability to obtain a large number of solutions in a short time.

The main approximation involved in the propagation theory is again that the errors propagate according to a set of linear differential equations. The approximation was justified by solving the linear error-propagation equations for a typical problem. The problem was of average size for the M.I.T. Flight Simulator. It involved 12 integrations and a number of nonlinear operations such as vector resolutions and multiplications. The solution to the linear error-propagation equations was obtained using the computer as set-up to solve the original problem. The solution was obtained in the form of a set of superposition integrals. With use of the superposition integrals, it was possible to calculate the errors caused by any component errors and to compare the calculated errors with actual errors observed in the solution. A good agreement between calculated and observed errors was obtained.

While the theory for determining the errors was justified, the practical difficulties in applying the theory were shown to be great. The impasse in all the error-analysis work done so far has not been the failure of or even the lack of theories, but rather the impossibility of applying these to achieve useful results with a reasonable amount of time and effort.

2.4. Practical Operational Techniques.

Two methods have been developed for the rapid discovery and elimination of faulty computer components and setup errors. The methods were evolved gradually by the operators of the M.I.T. Flight Simulator; hence, the work done under the error-analysis program consisted mainly in evaluating and describing the operators' techniques. The two methods discussed in this report are general in nature and can be applied to a wide variety of analogue computers, not merely to computers similar to the Simulator.

The first of these methods, "Static Checking," effectively eliminates all the time-varying elements in the computer and allows the constants of the nondynamic elements to be measured accurately. A "Static Check" is a relatively simple, fast means of assuring the proper operation and connection of a large portion of the computer. Of course, the "Static Check" can give no information concerning the dynamic computer elements.

In order to test the dynamic elements, a second method called "Dynamic Checking," was developed. A "Dynamic Check" exercises and examines both the static and dynamic elements of the computer. "Dynamic Checks" range from very simple tests of single computing elements to complete tests of the operation of the entire computer. In general, the "Dynamic Checks" are more complicated than the "Static Checks" and require more advanced preparation.

In addition to developing these trouble-shooting methods, the work of evaluating actual computer operating procedures showed that the physical arrangement of the computer has an important effect on the time needed for trouble shooting. In particular, it was concluded that a great deal of time could be saved merely by the construction of certain auxiliary interconnection panels. Such auxiliary panels decrease trouble-shooting time by providing rapid access to all the signals in the computer.

2.5. Flight Table Report.

In April, 1951, a comprehensive analysis and evaluation program was initiated to obtain positive information on the sources of errors in the 3-axis flight table associated with the M.I.T. Flight Simulator. Known sources of errors included drift, backlash, friction, saturation, and dynamic effects associated with multigimbal operation. As a background for this program, a report was to be prepared summarizing the status of the Flight Table at that time because no such compilation of information was available.

The preparation of the report, "Interim Report on the Flight Table Section of the M.I.T. Flight Simulator," has proven to be a more extensive task than first visualized. To date, a large portion of eight of the ten chapters has been written, and the completed report may require between 400 and 500 pages in two volumes, the larger one unclassified, and the smaller classified SECRET in order to include the results of certain simulator programs. Every effort is being made to present a complete, accurate description and evaluation of the Flight Table, with associated components, and to draw from this information positive conclusions that may serve as guides in future work. In order that the material which has been written may be of immediate use, it is being given a very limited advanced distribution in hectograph form.

Concurrent with the preparation of the report, a number of tests have been conducted to isolate sources of drift and nonlinear operation in the servos associated with the Flight Table. These tests utilized elementary methods to give information that will be helpful in improving servo performance before a more detailed analysis is made by means of methods discussed elsewhere in this memorandum.

3. THE STATISTICAL EVALUATION OF ERRORS GENERATED BY COMPUTER COMPONENTS

The work summarized in this section as directly applied to computer components is taken from Mathews' thesis.[1] However, in as much as most of the analogue computer components are typical of components used in many other systems, the component-evaluation methods here presented have wide applications. Some methods that were developed for evaluating the performance of computer components have far outgrown their original area of usefulness and are being applied to a variety of control systems. Advanced research of this type is of a more general nature and is not of direct application to the error-analysis program. Consequently, some of the advanced research has been separated from the error-analysis program and is being done as part of another program Some of this research is published in two D.A.C.L. reports.[7, 8]

The two problems now to be considered are the following:

1. Evaluation of the transfer characteristics of a component.
2. Representation of component errors as extraneous sources adding errors to the signals in the component.

In both problems, a statistical approach is taken. Methods are developed for estimating the average values of various errors. The statistical approach is necessary, partly because it is the most realistic way of expressing certain inherently random errors and partly because more specific methods of analysis did not prove feasible because of the lengthy computations involved.

3.1. Evaluation of Components.

Most computer components are part of a class of apparatus designed to operate linearly but in which significant operating errors are caused by inescapable nonlinearities. Even nonlinear components such as multipliers can be included in this class because the nonlinear operation is achieved by an assemblage of parts, most of which are intended to operate linearly. Linear theory is usually sufficient to determine the gross way in which the components function. However, in determining the deviations between how the component works and how it should work, the nonlinearities become very important and must be taken into account.

Two related types of problems arise in the evaluation of a component. First, the linear and nonlinear characteristics of the components must be determined. Second, once the character of the component is known, the errors generated when a given type of input is applied to it must be calculated. Numerous attempts have been made to solve the first problem by dismantling the component either actually or figuratively, examining the characteristics of the individual parts and then reassembling the individual characteristics to see how the component operates. Usually the attempts have succeeded only partially, probably because it is not possible to take proper account of the nonlinearities in the parts. A synthesis process in which the composition of a component is deduced from its response to suitably chosen tests often proves more satisfactory. The second problem is one of analysis because here the system and the input are given and the response is to be determined.

For lumped-constant, time-invariant linear systems, fairly general methods have been developed for both the analysis and the synthesis problems. As might be suspected, the synthesis problem is more difficult than the analysis problem. However, for nonlinear systems there are no general methods for solving either problem. In consequence, special techniques have been developed for handling limited classes of nonlinear systems. At the present time, the analysis problem has been solved more completely than the synthesis problem. A method has been developed at the D.A.C.L. for determining the approximate statistical response of a class of nonlinear systems to Gaussian random input signals. The class includes systems made up of (1) linear elements, either passive or active, with energy storage; (2) feedback paths; and (3) nonlinear elements without energy storage, with an output which can be expressed as an instantaneous function of input. Thus, the class includes a great number of computing elements with significant nonlinearities such as limiting and backlash. Therefore, even though the analysis is not general, it is of great practical value.

The inputs considered are Gaussian random signals. Similar methods of analysis have been developed where the inputs are sinusoids.[9, 10] However, the random-signal analysis possesses several advantages which make it of great interest. The signals in computers are usually not sinusoids, thus the sinusoidal analysis is not especially appropriate. Furthermore, the exact nature of the signals is seldom known in advance, and it is necessary to characterize the signals statistically. Also a statistical characterization allows the rapid calculation of over-all average quantities and affords a great saving in time over calculating the exact outputs from specific inputs. The analysis scheme considers only Gaussian random signals, partly because these are the only type of random signals for which it is as yet feasible to carry out calculations, and partly because such signals are a realistic approximation to many computer signals.

It is beyond the scope of this report to describe the details of random-signal analysis procedure. References are given to two D.A.C.L. reports.[7, 8] One example of the results of an analysis applied to a typical computing element is presented to show the general type of information obtained. The computing element is the electromechanical integrator servo used in the M.I.T. Flight Simulator. A block diagram of the servo is shown in Fig. 1. The integrator is basically a rate servo using tachometer feedback to achieve an approximate integration between an electrical input and mechanical output.

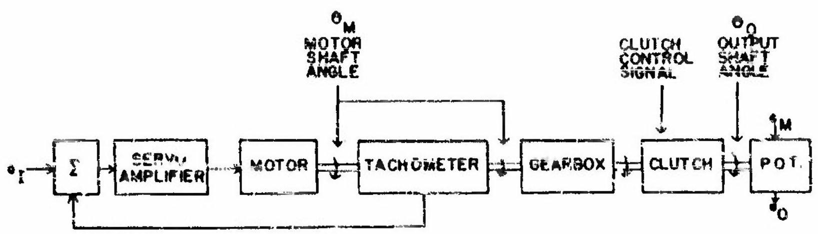
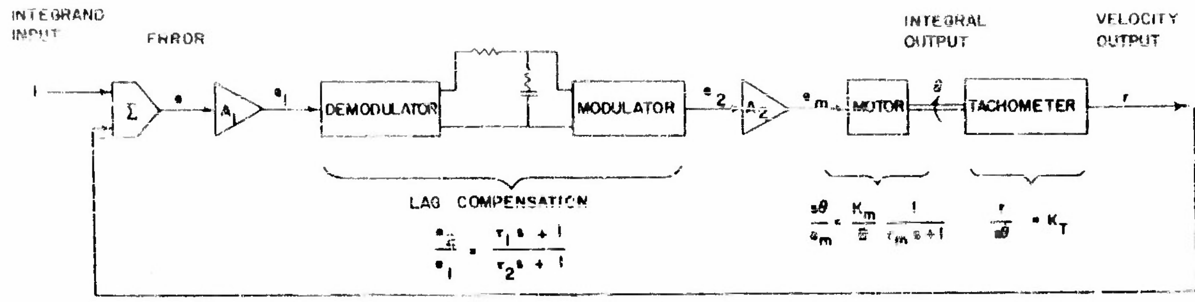


Fig. 1. Block diagram of integrator servo.

The mechanical output is coupled through a clutch and gear train to a potentiometer which converts the mechanical output to an electrical signal e_O which is proportional to the mechanical output times the voltage e_M which excites the potentiometer. The analysis takes into account the primary feedback loop including the servo amplifier, motor, and tachometer. The errors introduced in the gearbox, clutch, and potentiometer are considered in Sec. 3.2. Figure 2 shows a detailed diagram of the feedback system and gives the time constants of the various parts of the system. The principal nonlinearity in the integrator is the torque limiting in the servomotor.

The response of the servo to a random input signal with a quadratic power spectrum is graphed in Fig. 3.[8*] The points are experimental results and the curves are analytic results. The response measured is the rms error E, because for error analysis this is more significant than the actual output R. The response is plotted as a function of the

*Figure 3 is taken from D.A.C.L. Report No. 70, in preparation.



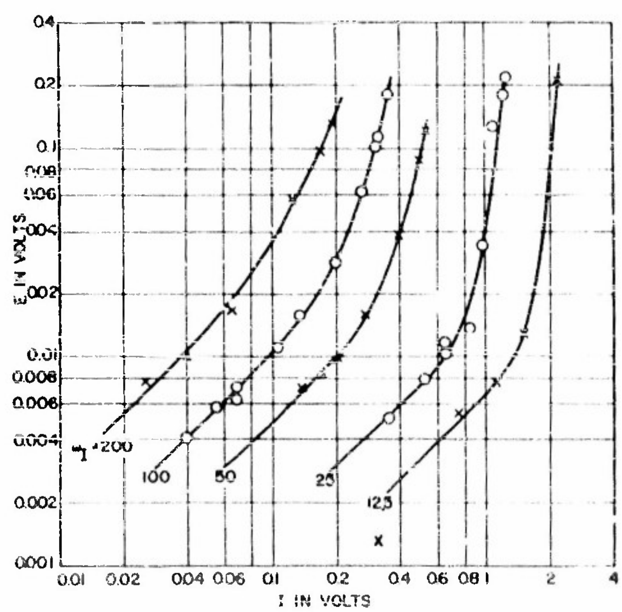
Compensation time constants $\begin{cases} \tau_1 = 0.008 \text{ sec} \\ \tau_2 = 0.136 \text{ sec} \end{cases}$

Motor-tachometer time constant $\tau_m = J/B = 0.75$ sec
where J is combined inertia of motor and tachometer
and B is combined viscous friction of motor and tachometer

Zero-frequency loop gain $K = (A_1 A_2 K_m K_T)/B = 9500$

Torque limit referred to the velocity output
$=$ Maximum motor torque x $(K_T/J) = 40$ volts/sec

Fig. 2. Detailed diagram of feedback system.



Power spectrum of the input

$$\approx \left| \frac{1}{\frac{s^2}{\omega_I^2} + \frac{2\zeta s}{\omega_I} + 1} \right|^2_{s = j\omega}$$

Fig. 3. Response of integrator servo.

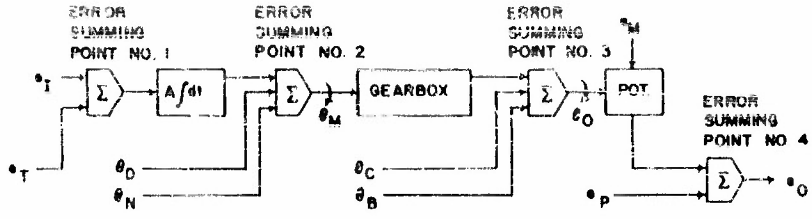
rms input I for various values of bandwidth of input ω_I. It is possible to see heuristically what the shape of the response curves should be and that three different regions of operation should exist. At low amplitudes of input, limiting is insignificant, and the error is a linear function of I for any particular value of ω_I. As I is increased, the effect of limiting causes the error to increase more rapidly and the E curve bends upward away from the linear portion. If the input was increased sufficiently, E again becomes a linear function of I, this time because E equals I. Here the servo no longer has any error-reducing effect. Figure 3 shows only the lower two operating regions since the experimental data were not extended to the region where E equals I. Linear calculations can be used to find the response in the region of low input, but a nonlinear analysis is necessary to calculate the response in the middle region. The generally good agreement between the experimental points and the analytic curves of Fig. 3 justifies the approximations made in the analysis for this particular case of a typical computer component.

The corresponding nonlinear synthesis problem can be presented in terms of the analysis problem just considered. The data for the synthesis problem are the response curves shown in Fig. 3 along with such information about the form of the component as would always be known in a practical problem. By form of the component is meant its operational structure. In the example the form is that of a single-loop feedback system. Probably something would also be known about the location and character of the nonlinearities. From these data, the various time constants and the character of the nonlinearity shown in the block diagram of Fig. 2 are to be determined. The procedure for the determination has not been worked out as completely as the analysis procedure. However, enough has been done to indicate the usefulness and importance of the synthesis method that is beginning to be developed. The work which has been published is in Mathews' thesis[1] and no attempt will be made to summarize it. Only two comments will be made. First, the synthesis is based on the same approximations that are involved in the analysis, and thus should be valid for the same type of system. Second, since the synthesis uses the response of the component as data, the synthesized system will respond in the same way as the actual component. The response is the most important aspect of the component for error analysis. Thus, the synthesis should prove to be an important tool for error analysis.

3.2. Representation of Component Errors.

After a component has been evaluated, the problem of how to represent the errors produced by the component still remains. Usually it is not convenient to consider a computer to be made up of a collection of real components with characteristics differing slightly from ideal components. The response of such a computer would be very difficult to determine because real components are much more complex than ideal components. Instead, it is better to consider the computer to be constructed from ideal components plus a number of error sources that produce an effect equivalent to the errors generated by the real components. The errors in the real computer can then be determined as the response of the ideal computer to the error sources. This determination is difficult, but is far easier than directly determining the response of the real computer.

Most real components can be represented in this manner, but a number of sources may be required for a good representation. As an example, an error diagram used for the integrator servo is illustrated in Fig. 4. Four error summing points are shown. At the first two summing points, errors arising from (1) the inexact transfer function of the real integrator e_T, (2) drift in the integrator θ_D, and (3) noise picked up in the integrator θ_N are added to the input and output of an ideal integrator. Each of these errors can be evaluated experimentally, at least for its statistical properties. The transfer–



e_I = Integrand input (volts)
θ_M = Motor shaft angle (radians)
θ_O = Output shaft angle (radians)
e_M = Multiplier input (volts)
e_O = Electrical output (volts)
A = Nominal integrating constant (rad/volt-sec)

e_T, θ_D, θ_N, θ_C, θ_B, e_P — Error Sources

Fig. 4. Integrator error diagram.

function errors e_T are essentially those shown on Fig. 3 for the case of a random intr-grand input. Other errors such as clutch slippage θ_C, backlash in the gear train θ_B, and error introduced by the potentiometer e_P are introduced at the last two summing points.

The definitions of these errors and the points at which they are added into the component signals are in no sense unique. The important consideration is to choose convenient definitions both with regard to the measurement of the errors and the effect of the errors in the component.

The problem of measuring errors is not trivial. The main difficulty is that computer components in many cases have been perfected to a higher degree than readily available measuring instruments. Thus, the errors in the components may be smaller than those in the measuring instrument. This difficulty is most severe for dynamic measurements. In consequence, indirect ways of measurement must be developed for many errors.

In the integrator servo example, the effect of all the error sources can be combined into one error added to the output of the component. The equivalent circuit of the real component is shown in Fig. 5 where the output error E_O may be evaluated from Fig. 4 and is

$$E_O = K_P\left[\left(A\int e_T dt + \theta_D + \theta_N\right)G + \theta_C + \theta_B\right] + e_P \qquad (1)$$

where K_P is the conversion constant of the potentiometer (volts/radian) and G is the gear ratio. The method of representing the errors in Fig. 5 is convenient for studying the propagation of errors in a computer, as will be shown in Sec. 5.

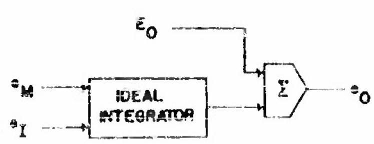


Fig. 5. Representation of real integrator.

Some of the error sources in the integrator servo, such as the noise error θ_N and the drift error θ_D are definitely random in nature and cannot be evaluated as specific time functions. Other error sources, such as transfer-function errors e_T are functions of the input to the servo, and could be evaluated as a specific time function for a specific input. Still other sources, such as backlash θ_B and potentiometer errors e_P, are in an intermediate group which depend somewhat on the input but also are affected by random variables. Even though it is possible to calculate e_T as a specific time function, such a computation is exceedingly long because the nonlinearities in the component require the computation to be made by some numerical-analysis method. In consequence, no attempt yet has been made to evaluate time-function errors. Instead, statistical errors have been evaluated for the case where the component input is a random signal, and thus where all the errors in the component are random. For a random input, the various error sources in Eq. (1) can be assumed to be uncorrelated. The mean-square output error $\overline{E_O^2}$ is then the sum of all the mean-square errors and can be written

$$\overline{E_O^2} = K_P^2\left[\left(\Lambda^2\int\overline{e_T^2}\,dt + \overline{\theta_D^2} + \overline{\theta_N^2}\right)G^2 + \overline{\theta_C^2} + \overline{\theta_B^2}\right] + \overline{e_P^2}\,. \tag{2}$$

The mean-square output error as given in Eq. (2) has been evaluated experimentally for the integrator servo, and is a measure of the quality of the servo. Not only does $\overline{E_O^2}$ provide a measure of the total error but also makes easy a comparison of the relative magnitudes of the various terms that make up $\overline{E_O^2}$, and thus a comparison of the relative importance of the various sources.

For a typical component, anything approaching a complete representation of the errors requires a number of error sources. The problem becomes involved, even when only statistical errors are considered, as can be seen from the complexity of Fig. 4 and Eqs. (1) and (2). Thus, for the large number of components in the usual computer problem, a statistical approach is the only one which could possibly produce an error-analysis problem of a manageable size.

4. ANALYTIC INVESTIGATION OF THE PROPAGATION OF ERRORS IN A SET OF DIFFERENTIAL EQUATIONS

4.1. The Problem of Error Propagation.

A knowledge of the way in which errors propagate in a problem consisting of a set of differential equations is of importance in estimating whether a problem can be solved on a given computer to a specified accuracy. How the errors propagate is a measure of the sensitivity of the problem to the errors that will always be introduced by the components of any actual computer. The sensitivity, however, is a function of the problem itself and not of the specific computer on which the problem is solved.

The work described in this section was done by H. Mori[2] in his thesis, "The Analysis of Numerical Check Solutions for the M.I.T. Flight Simulator." The particular thesis problem was the examination of the propagation of errors in a numerical check solution. Although Mori actually dealt with numerically solved check solutions, his work in calculating error sensitivities is of direct interest in the error-analysis program because the sensitivity of a problem is independent of how the problem is solved and because his check-solution problems are typical analogue-computer problems. Also, since Mori determined error sensitivities analytically, without the assistance of an analogue computer, the methods he used might make it feasible to estimate the errors in a problem before it was actually solved on a computer— a procedure which is one of the objectives of the error-analysis program.

The meaning of the phrase "propagation of errors" should be clarified before going into the details of how the propagation is evaluated. It is convenient and theoretically always possible to present ordinary differential equations in the form

$$\frac{dZ_k}{dt} = f_k(Z_1, Z_2, \ldots, Z_n, t) \qquad k = 1, 2, \ldots, n. \tag{3}$$

The f_k functions are in the most general case nonlinear functions. The independent variable t in the differential equations may be thought of as being time without any loss of generality, and will be so referred to in the remainder of this report. The solutions to the differential equations are functions of time and may be written

$$Z_k(t) = S_k(t), \qquad k = 1, 2, \ldots, n \tag{4}$$

where $S_k(t)$ is the exact solution to the equations for a given set of initial conditions. The solutions depend not only on the f_k functions but also on the initial conditions which for equations in the form of Eqs. (3) are the values of $Z_k(0)$.

In solving the equations an error might be made in one of the variables at some time t_n. The way the error is actually introduced is not important since attention here is concentrated on what happens to errors once they are introduced. The assumption is made that

$$Z_k(t_n) = S_k(t_n) + E \tag{5}$$

where E is the introduced error. Introduction of the error is equivalent to beginning a new problem at time t_n with initial conditions slightly different from those of the original problem. The error in the general case will change all the $Z_k(t)$ functions for $t>t_n$. Thus it can be said that the error introduced at one point will propagate through the rest of the solution. As t increases, the error E may either decrease or increase depending on the nature of the f_k functions. The case in which the error increases is of course the most difficult for computation, and such problems can be called sensitive to errors. The rate of increase is a measure of the sensitivity, and the task for error analysis is to find some way of evaluating the sensitivity.

The previous discussion has considered only the effects on all the variables of an error committed in one variable at one time. In an actual problem the effects of an error committed at any time on any variable must be considered. Thus a problem with n variables, in which it is desired to examine the error characteristics at m different times, will have $n^2 \times m$ sensitivity functions. Clearly, even a small problem will require a great deal of information to specify its sensitivity, and some way of simplifying the requirements is essential.

4.2. Linearized Study of Error Propagation.

Because the introduction of errors in a problem is equivalent to a change in initial conditions for the problem, the propagation is governed by the differential equations of the problem. If these equations are nonlinear, then the errors propagate according to a set of nonlinear equations. Although it is possible to solve the nonlinear equations, no general methods exist for simplifying and extending the solutions. The error sensitivity of the problem always depends on the specific errors made during the solution, and no means is available for separating the sensitivity of the problem from the exact method by which it is solved. In view of the large amount of information required to specify the error sensitivity of a problem, some way to simplify and generalize the sensitivities is a practical necessity.

The method used to achieve the simplification is to assume that the errors are sufficiently small with respect to the problem variables that they propagate according to a set of linear differential equations. The many general mathematical tools for analyzing linear systems then are available to study the errors. In particular, superposition is applicable, and the separate effects of individual errors applied one at a time can be summed to obtain total solution errors. It is thus possible to make a practical separation between the sensitivity of a problem to errors and the specific errors committed during its solution. Of course, linearizing the error-propagation equations is admittedly an approximation, but it is certainly a necessary approximation in view of the complexity of the error-analysis problem. Mori's work in showing the validity of the linearization is, therefore, of considerable significance.

The approximation procedure used to linearize the error equations is a fairly standard process. The particular methods used by Mori were presented by Brock and Murray[5] in a Project Cyclone report. Slightly different methods have been developed by Jones,[6] and these will be discussed further in Sec. 5 which deals with the propagation of errors in actual computers. The linearization method used by Mori can be illustrated in terms of the problem specified by Eqs. (3). If the equations are solved by some inexact method, then the approximate solution $Z_k(t)$ can be expressed as

$$Z_k(t) = S_k(t) + E_k(t), \qquad k = 1, 2, \ldots, n \tag{6}$$

where $S_k(t)$ is the exact solution and $E_k(t)$ is the error. After substitution of Eqs. (6) into Eqs. (3) and the assumption that $E_k(t)$ is sufficiently smaller than $S_k(t)$, then it can be assumed that

$$\frac{dS_k}{dt} + \frac{dE_k}{dt} = f_k(S_1, S_2, \ldots S_n, t) + \sum_{j=1}^{n} \frac{\partial f_k}{\partial S_j} E_j(t) \qquad k = 1, 2, \ldots, n \tag{7}$$

without the introduction of a significant error. But since $S_k(t)$ is the solution to Eqs. (3),

$$\frac{dS_k}{dt} = f_k(S_1, S_2, \ldots S_n, t) \qquad k = 1, 2, \ldots, n, \tag{8}$$

and Eqs. (8) may be subtracted from Eqs. (7) with the remainder

$$\frac{dE_k}{dt} = \sum_{j=1}^{n} \frac{\partial f_k}{\partial S_j} E_j(t) \qquad k = 1, 2, \ldots, n\,. \tag{9}$$

Equations (9) are the desired set of linear equations governing the propagation of errors. Actually Eqs. (9) are the homogeneous equations for the propagation. No mention has been made of a forcing function, consisting of the specific errors made during the inexact solution of the original differential equations. The solutions to the homogenous equations will specify completely how the errors propagate and it is not difficult to determine any particular solution from the homogenous solutions.

Although Eqs. (9) are linear, they are not constant-coefficient linear equations. The coefficients $\partial f_k / \partial S_j$ vary with time not only because they may be direct functions of time but also because they are functions of the Z_k's which vary with time. As a consequence, exact solutions for the equations cannot be obtained in terms of elementary functions. In order to avoid solving the equations numerically, a procedure that would have taken a great deal of computation, Mori divided the equations into sections in time and approximated each section by a set of constant-coefficient equations. He was thus able to obtain an approximate solution to the equations in terms of sections of analytic functions. Theoretically, by using many small sections, a very good approximation to the solution could be obtained. However, to save work, Mori used only a few large sections. By this fairly crude method he was able to obtain an accuracy sufficient for error analysis. Nevertheless, it should be pointed out that there are two essential approximations in his final error-propagation solutions. The first is the linearization of the original equations, and the second is the approximation of the linear equations by piecewise constant-coefficient equations.

Most of the time spent in evaluating the error propagation in the problems that Mori considered was used to solve the piecewise constant-coefficient equations. The details of solving the equations are straightforward and are not of sufficient interest to be included here. Matrix methods were used to allow orderly calculations and to obtain the required general homogeneous solutions to the equations with all arbitrary constants. The problem considered reduced to four independent variables; hence, 4 × 4 matrices were involved. The main work of the solution consisted of finding the characteristic roots and characteristic vectors for the matrices corresponding to each of the piecewise constant-coefficient sections. As a typical example of the amount of time involved, approximately 70 man-hours of work were required to solve one problem that was divided into three sections. This is approximately the same time as required to obtain a complete solution to the original problem by numerical-integration methods using a desk calculator.

4.3. Practicability of Analysis of Propagation of Errors in Differential Equations.

The particular check solution that was examined as a sample problem was chosen partly because two solutions to the problem were available, one of very high accuracy, the other of rather poor accuracy. Thus, the errors in the second solution could be compared with errors calculated in an error-propagation analysis, providing a check on the propagation analysis.

The problem, put in the form of Eqs. (3), is the following:

$$\frac{dY}{dt} = A \sin \theta + B \sin kt \tag{10}$$

$$\frac{dX}{dt} = A \cos \theta + B \cos kt \tag{11}$$

$$\frac{d\theta}{dt} = \psi \tag{12}$$

$$\frac{d\psi}{dt} = f(X, Y, \theta, \psi) \tag{13}$$

where X, Y, θ, and ψ are the dependent variables and t is the independent variable. The problem solution is of interest for t between zero and T. The linearized error-propagation equations corresponding to Eqs. (9) are

$$\frac{dE_Y}{dt} = A(\cos \theta)E_\theta \tag{14}$$

$$\frac{dE_X}{dt} = -A(\sin \theta)E_\theta \tag{15}$$

$$\frac{dE_\theta}{dt} = E_\psi \tag{16}$$

$$\frac{dE_\psi}{dt} = \frac{\partial f}{\partial Y} E_Y + \frac{\partial f}{\partial X} E_X + \frac{\partial f}{\partial \theta} E_\theta + \frac{\partial f}{\partial \psi} E_\psi \tag{17}$$

where E_Y, E_X, E_θ, and E_ψ are the respective errors in Y, X, θ, and ψ. The coefficients in Eqs. (14) through (17) are functions of time. For example, in Eq. (14) A sin θ varies with time. By an examination of the variation of the coefficients, the problem was roughly sectioned into three parts, 0 <t<0.4T, 0.4T<t<0.8T, and 0.8T<t<T. Figure 6[2*] shows a typical solution to one of the linearized sections of Eqs. (14) through (17). Here the error propagated in X caused by an error injected into X is graphed. In this case the propagated error decreases with time, demonstrating that this particular portion of the problem is not very sensitive. Naturally, many other propagation functions like Fig. 6 would have to be shown to specify completely the error sensitivity.

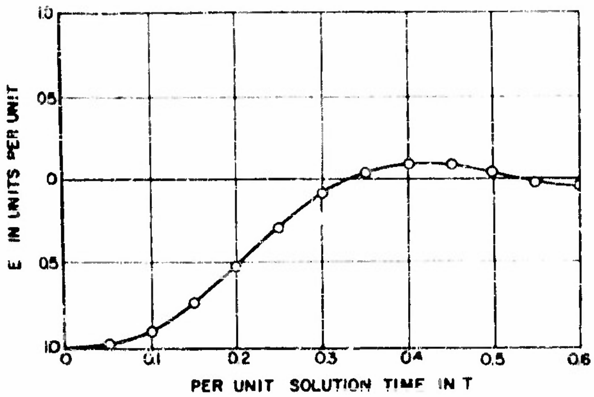


Fig. 6. Error in X caused by unit error injected into X.

The over-all error in X caused by all the computational errors is shown in Fig. 7.[2†] The calculation of the error in X required estimating the errors committed during the solution of the check solution. However, because the check solution was solved numerically, this estimation was possible. In Fig. 7 the calculated errors are compared with the observed errors. The calculated errors agree very well with the observed errors for small values of t, and for all values of t the calculated errors show the order of magnitude of the observed errors. The apparent discontinuity in the observed

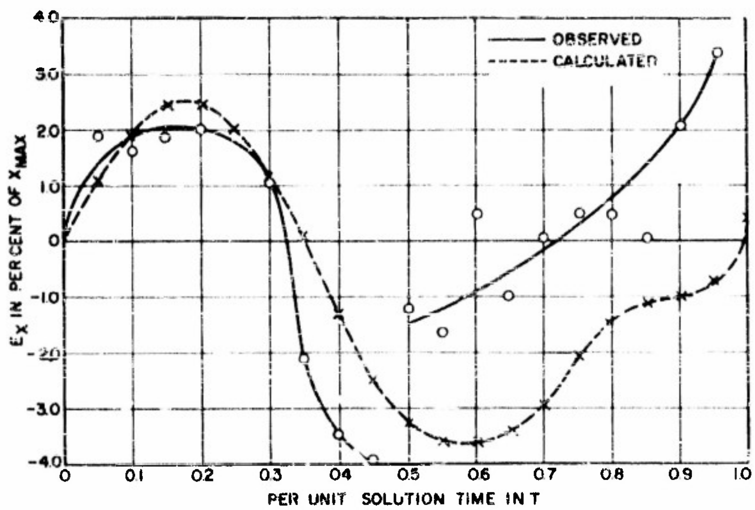


Fig. 7. Observed and calculated E_X.

*Figure 6 is adapted from Fig. 3.3 of Mori's thesis.

†Figure 7 is adapted from Fig. 3.11 of Mori's thesis.

error was due to unusual addition of the round-off errors made during the numerical solution of the check solution. This round-off error was not taken into account in the calculated error response. Had the round-off error been taken into account, a much better agreement between the two error curves would have been obtained.

The accuracy of the calculated error curve is certainly as good as is required to estimate error magnitudes in practical applications. Thus the approximations made in linearizing the error-propagation equations and in sectioning the time-varying linear equations are justified.

Several conclusions can be drawn from the analysis and examples of error propagation in a problem. A good estimate has been obtained of the requirements for determining the error sensitivity of a problem and for predicting the errors that will be made by solving the problem with some actual computer.

In order to calculate the error sensitivity, it is necessary to have, in addition to the differential equations for the problem, an approximate solution to the problem. Knowledge of the approximate solution is necessary in order to evaluate the $\partial f_k / \partial Z_j$ coefficients in the error-propagation equation. However, as was shown by the success of the fairly crude approximations used in the example, only a rough idea of the solution is necessary. The time required to calculate the sensitivity depends on the size of the problem, the size being measured by the number of variables. A four-variable problem took approximately 70 man-hours to compute. The amount of work required to manipulate a matrix varies approximately as the factorial of the size of the matrix; hence, the time required increases very rapidly as the number of variables is increased. Mori is of the opinion that sensitivities of problems with more than five variables will have to be calculated with the help of an automatic digital computer because hand calculations would be too tedious. Usually, it would be desirable to make rapid estimates of a problem sensitivity prior to its acceptance for study on an analogue computer. Making rapid estimates would require the immediate availability of a digital computer, preferably in the same laboratory with the analogue computer. The combination of digital and analogue computing facilities might be very effective for studying a number of kinds of problems. However, such a combination would be too expensive for a small organization to maintain.

A knowledge of the error sensitivity of a problem may be sufficient to evaluate its computability. However, if it is desired to estimate the errors that will be made on the problem with a given computer, then the actual time-function errors introduced by the computer components must be known as well as the problem error sensitivities. As was pointed out in Sec. 3.2, calculation of the time-function errors is not feasible because it takes too long. Also if the linearity and piecewise constant-coefficient assumptions are allowable, determination of the propagated errors from the injected errors is a long computation. The over-all conclusion is that it is not worthwhile attempting to estimate specific propagated errors analytically by the methods of this section. On the other hand, the estimation of error sensitivities may or may not be justified depending on the specific problem at hand and the facilities available to make such an estimate.

5. EXPERIMENTAL INVESTIGATION OF THE PROPAGATION OF ERRORS IN A COMPUTER

As part of the error-analysis study, the actual errors in a computer solving a typical problem were to be investigated experimentally. The general purpose of the investigation was to develop practical operational techniques for estimating the errors in the computer solutions. The line of attack on the problem was to measure the sensitivity of the problem to errors of the type that are produced by the computer components and to correlate this sensitivity with the actual errors observed in the computer solution. In particular, it was desired to determine the amount of computing time required to obtain sufficient error-sensitivity data to estimate with confidence the errors in a particular computer solution or set of solutions. The major part of this particular investigation was carried out by Rabow, and the details of this work are reported in his thesis.[3]

5.1. Linearization of Propagation Analysis.

In order to analyze the propagation of errors in a computer, it is necessary to assume that the errors are relatively small and thus that they propagate according to a set of linear differential equations. A similar linearity assumption was required in Sec. 4.2 dealing with the propagation of errors in a set of differential equations. Methods also were given in Sec. 4.2 for finding the linear differential propagation equations and for obtaining approximate analytic solutions of them. If the errors are to be determined experimentally, it is not necessary to exhibit the error equations because they can be solved experimentally on the computer as set up to solve the original problem for which the solution errors are to be determined. However, it is essential to know that the propagation equations are linear, and thus to be able to specify the form of the solution. Also, while in Sec. 4.2 solutions of the homogeneous propagation equations were of the greatest interest in specifying the error sensitivity of the problem, here the main interest is centered on particular solutions of the propagation equations. As a consequence, the ways in which errors are committed in actual computers must be considered. Errors will be referred to as being injected by the computer. The precise meaning of injected errors will be made clear subsequently.

The particular solution to a linear differential equation can always be expressed as a superposition integral. Where the equations do not have constant coefficients, the kernel of the integral is a general function of two variables. The method of expressing the error propagation in terms of superposition integrals was used by Jones.[6] An error in a dependent variable of computation can be written

$$E_{pq}(t) = \int_0^t \lambda_{pq}(t, \tau) e_q(\tau) d\tau. \tag{18}$$

The quantity $E_{pq}(t)$ is the error in the variable p caused by an error $e_q(t)$ injected into variable q. Both $E_{pq}(t)$ and $e_q(t)$ are functions of time. The kernel, $\lambda_{pq}(t, \tau)$, can be considered to represent the response at time t to a unit impulse error injected at time τ. The total error in variable p is the sum of the errors produced by errors injected in all the other variables. Thus, in order to specify the errors in all the variables in an n variable problem, n^2 kernels must be determined. In the computer, $\lambda_{pq}(t, \tau)$ can be measured as a continuous function of t, for discrete values of τ. If m values of τ are considered, then $m \times n^2$ functions are required to specify completely the error propagation in the problem. Thus, the same amount of information is required here as was required to specify the problem error sensitivity in Sec. 4.1. Indeed, Mori[2] has shown that it is possible, though impractical, to calculate the superposition integral kernels from the error sensitivities. The experimental methods developed by Rabow[3] for evaluating the kernels are more practical.

It is important to define injected errors precisely and to distinguish between the error injected into a computer variable and the resulting error caused by the injected error. Although injection could be defined in a number of ways, the most convenient definition found so far is to consider the injected error to be added to the variable it affects. Thus, for example, in Fig. 8 the variable $Z_k(t)$ is generated at some point, a, in a computer and is transmitted to other points, b, c, and d, in the computer. To inject an error e_k) into $Z_k(t)$, the $Z_k(t)$ channel is broken, and a summing circuit is inserted, as shown. For the example, $e_k(t)$ can be considered the only error injected into the computer. If $e_k(t) = 0$, then $Z_k(t)$ equals $S_k(t)$, the exact solution to the problem. For $e_k(t) \neq 0$, $Z_k(t)$ will equal $S_k(t)$ plus some error $E_k(t)$ that is produced by the injected error, $e_k(t)$. It can be seen that unless the computer contains no feedback loops between b, c, or d, and a, $e_k(t) \neq E_k(t)$, and thus the injected error will not equal the resulting error. The usual computer problem requires feedback loops. In this case $E_k(t)$ must be determined from $e_k(t)$ by means of a superposition integral with $\lambda_{kk}(t, \tau)$ for a kernel.

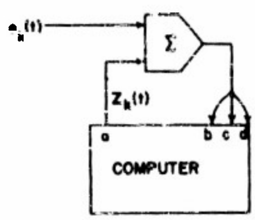


Fig. 8. Error injection by addition.

There are two principal advantages to defining error injection by a summation. First, as was pointed out in Sec. 3.2, it is possible to represent component errors as extraneous variables that are summed with the output of the component. Thus component errors can be treated directly as injected errors. Second, since summing circuits are readily available in most computers, it is possible actually to insert summing circuits in the channels of the variables and, by injecting artificial errors, to evaluate the kernels of the superposition integrals.

5.2. Experimental Evaluation of Superposition-Integral Kernels.

An experimental evaluation of the superposition-integral kernels was carried out for a typical problem. The same computer setup was used to solve the problem and to evaluate the kernels, except for addition of some summing circuits in the latter process to facilitate the injection of artificial errors into the computer. The use of essentially the same setup for solving the problem and for evaluating the error kernels is necessary if the evaluation of the error kernels is to be a feasible operational method for estimating solution errors while the problem is being solved.

The kernels were evaluated by obtaining the response to artificially injected errors. The response and the injected error were used in equations of a type similar to Eq. (18) to solve for the kernel, $\lambda_{pq}(t, \tau)$. Injection was accomplished by means of summing circuits, as illustrated in Fig. 8. There is nothing in Eq. (18) that restricts the type of injected error $e_q(t)$, which may be used to evaluate $\lambda_{pq}(t, \tau)$. However, $e_q(t)$ must satisfy certain other requirements, both of a theoretical and of a practical nature. The kernel $\lambda_{pq}(t, \tau)$ could be evaluated most simply by injecting an impulse error into the q variable at time τ_1, since for these conditions the resulting error $E_{pq}(t)$ would equal $\lambda_{pq}(t, \tau_1)$. Because linearity requires that the errors be small, it is necessary to approximate the impulse error by a pulse of finite height and width. The response to the finite pulses approximates $\lambda_{pq}(t, \tau)$. Usually a fairly critical compromise must be made between making the amplitude of the pulses too large, thus violating linearity requirements, and making the pulses too small, thus producing an error too small to measure. Only when the compromise can be made, can the kernels be evaluated with pulse injections.

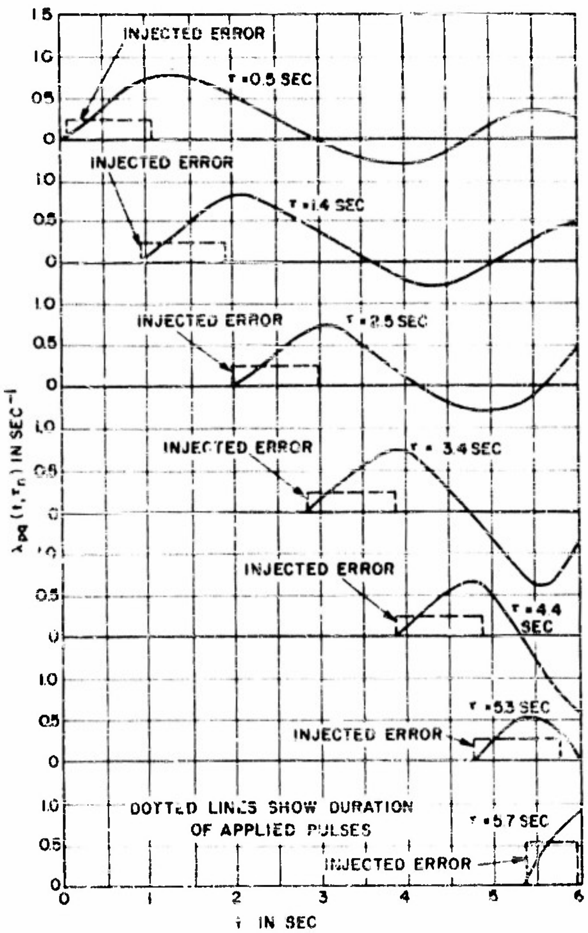


Fig. 9. Error-response kernels, $\lambda_{pq}(t, \tau_n)$.

Figure 9* graphs a typical set of pulse responses. The pulses are injected at various times during the solution, as shown by the dotted rectangles. The resulting errors are indicated by the solid curves. The seven solid curves of Fig. 9 comprise a family of $\lambda_{pq}(t, \tau_n)$ curves plotted as functions of t for seven values of τ_n. The value of τ_n can be taken as the mid-point of the time during which the pulse is applied.

The use of injected error functions other than pulses was tried. It was found that step-function injected errors were usually more satisfactory than pulses because they were easier to generate experimentally, and because it was easier to obtain a

*Figure 9 is adapted from Fig. 2-1 of Rabow's thesis.

resulting error large enough to measure without exceeding linearity limitations. The kernel, $\lambda_{pq}(t, \tau)$ of Eq. (18) can be calculated by differentiating the response to a step injected error, or a different superposition integral may be considered.

$$E_{pq}(t) = e_q(0)\Gamma_{pq}(t, 0) + \int_0^t \Gamma_{pq}(t, \tau) \frac{de_q(\tau)}{d\tau} d\tau. \tag{19}$$

For a superposition integral of the type of Eq. (19), the kernel $\Gamma_{pq}(t, \tau_1)$ is obtained directly as the response $E_{pq}(t)$ to a step-function error applied at time τ_1.

With either type of superposition integral, the errors produced by component malfunctioning can be calculated. The errors in the p channel produced by a simple gain error in the q channel were so calculated. Figure 10* shows the calculated error compared with the error actually observed in the computer with the q channel gain purposely misadjusted. The example shown in Fig. 10 is of necessity a simple one because the errors injected by the faulty component must be of such a type that they can be calculated in a reasonable time. However, the conclusions from the simple example apply to the more complex case. The close agreement between the calculated and measured errors shows the validity both of the linearity approximations made in the error-propagation analysis and of the methods used to evaluate the superposition kernels.

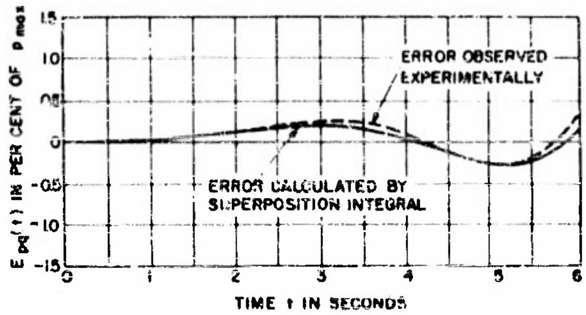


Fig. 10. Error in p caused by a gain error in the q channel.

5.3. Practicability of Analysis of Propagation of Errors in a Computer.

The agreement between calculated and observed errors in Fig. 10 has shown that if the errors generated by computer components are known, it is possible to calculate the errors produced in the computer solutions from the superposition-integral kernels. However, calculation of the errors does not appear to be feasible in most practical problems for three reasons. First, as has been pointed out in the last two sections, it is not feasible to find the actual time-function errors generated by components. Second, for a problem of reasonable size, measuring all the $n^2 \times m$ kernels required to describe completely the error propagation takes excessive computer time. Third, even after values of the kernels and the injected errors have been obtained, the actual computations to procure the resulting errors are long.

Instead of trying to calculate actual solution errors, it may be possible to obtain some information concerning the magnitude of solution errors for a particular computer

*Figure 10 is adapted from Fig. 2-2b of Rabow's thesis.

from the sensitivity of the problem to errors as expressed by the superposition-integral kernels. The assumption involved is that the components of the computer will always generate the same magnitude of errors; hence, the resulting solution errors will be proportional to the magnitude of the kernels. Such an assumption is crude and would have to be justified experimentally rather than being proved analytically. With such an assumption, it could be predicted that a computer which produced acceptably accurate solutions to one problem would produce solutions of similar accuracy to another problem with a similar sensitivity. The use of such an assumption appears to be the only reasonable way of estimating the magnitude of the solution errors.

The use of such an assumption still requires experimental evaluation of all the superposition-integral kernels. However, if something is known in advance about the error sensitivity of the problem, it might be possible to reduce the amount of experimental data required. The advance knowledge perhaps would be available from experience in working similar problems. By learning from past experience which kernels were especially large, a good idea of the error sensitivity could be obtained by evaluating only these critical kernels. Thus the work required to estimate the solution errors would be significantly reduced.

One important limitation to the process of estimating errors from the experimentally determined error sensitivities must be emphasized. In applying a linear analysis, the a priori assumption is made that the errors are small. If the computer produces a completely incorrect solution, the sensitivity measurements give no indication that the solution is incorrect. The sensitivities measured for the incorrect solution may show that the solution is relatively insensitive to component errors, and give no indication that large errors have been committed by the computer. An example of such behavior occurs when a computer solution becomes unstable through the influence of small extraneous time lags in the dynamic computing elements. The true solution to the problem in this case is not unstable, and will differ greatly from the computer solution. The sensitivity analysis can not show whether the instability is caused by computer errors or whether the original problem is unstable.

6. PRACTICAL TECHNIQUES FOR DETECTING AND ELIMINATING ERRORS IN COMPUTERS

6.1. Importance of Practical Techniques.

As mentioned in Sec. 2, experience with the M.I.T. Flight Simulator has shown that operational troubles can impair the efficiency of a computer. Although originally it was not thought worth while to study these practical problems in the error-analysis program, they have proved to be actually more important than some of the problems originally included, particularly when importance is measured in terms of computing cost. The practical troubles are best described from the point of view of an operator running a large computer. Since the operation of the computer is complex, the operator may have little understanding of whether or not the computer is functioning correctly, and even if he knows the computer is malfunctioning, he has little intuitive idea of the source of the trouble. Difficulties arise both from human mistakes and component failure. In a complicated problem, both the probability of human error and the probability of component failure will be accordingly large. Thus, in the course of a problem, and more particularly at the beginning of the problem, a number of troubles that must be eliminated before problems can be successfully solved are almost certain to arise. The operator's job is to eliminate the difficulties as quickly as possible. Fast elimination of troubles requires a rapid method for localizing the mistake or the malfunctioning component. In addition, the probability of making a second mistake while correcting the original one must be kept low. A number of rather general trouble-shooting techniques that meet these requirements have been developed at the D.A.C.L. The techniques are applicable to a wide class of analogue computers, and thus it is believed they represent a significant contribution to computer operating methods.

As expected, most of the trouble-shooting techniques have been developed by people closely connected with computer operations. The techniques summarized in this report are presented in a thesis[4] by Trembath. The subject of Trembath's thesis is the simulation of three-dimensional flight paths. Such simulation is one of the most complex computer problems studied on the M.I.T. Flight Simulator to the present time, and in Trembath's work, Sec. 3 on errors and methods of checking computers is an excellent presentation of trouble-shooting techniques, especially as applied to a large, complex problem.

6.2. Static Checking Methods.

One of the simplest methods for assuring the proper operation of a number of computer components as well as for localizing troubles is a static gain check. Computer components can be divided into static and dynamic elements. A typical example of the former is an amplifier with an output that is merely a constant times its instantaneous input, and a typical example of the latter is an integrator where the output changes with time, even for a constant input. By eliminating the dynamic elements in a particular computer setup, it is possible to make a slow and careful check of the proper operation of the static elements. For instance, in a computer consisting entirely of integrators, summing circuits, and amplifiers, by fixing the integrator outputs at predetermined values and by measuring the signals throughout the system, it is possible to deduce whether all the summing circuits and amplifiers are functioning correctly. The gains of either individual elements or groups of elements may be measured and compared with predetermined correct values. Thus a systematic method is provided for localizing errors by gradually decreasing the number of elements in the group under test.

In making static checks, it is important to have convenient check points that allow the rapid measurements of the signals throughout the computer without the necessity for disturbing the computer setup. In many cases, the ordinary interconnection panel or patch panel on which the various computing elements are interconnected does not form a satisfactory set of test points for checking purposes. Usually, the connection points on the panel are somewhat obscured by the interconnecting wires which cover the panel. Also, in many cases, it is necessary to break into a given signal channel and insert a multiple connector in order to measure the signal in the channel. Disturbing the patch panel tends to introduce further mistakes into the computer setup, and thus in a complicated problem, the probability of creating a new error may be almost as high as the probability of finding an existing error. To overcome this difficulty, duplicate patch panels have been installed on the M.I.T. Flight Simulator. All computer signals have been brought out to the duplicate patch-panel terminals which are used solely for testing. The auxiliary terminals not only make the testing process faster and more convenient but also eliminate errors introduced by disturbing the original wiring.

In addition to locating gross errors, it has been found that a complete static check performed periodically assists in locating components that are gradually developing errors. Thus, a static check is a valuable tool in maintaining the highest possible accuracy in the system.

6.3. Dynamic Checking Methods.

The value of the static checking methods is limited by the fact that neither the dynamic computer components nor the initial conditions are checked, and thus errors from such sources as malfunctioning integrators cannot be detected. Furthermore, despite the work reported in Secs. 4 and 5, the correlation between the static accuracy and the errors in the solutions is unknown. It is not possible to specify the allowable static-accuracy limits with respect to allowable solution errors. Therefore, dynamic checking methods must be developed to test the computer more completely. A number of these methods have been devised, ranging from very simple measurements of the time response of single components to determining the response of the over-all computer system as set up to solve a specific problem. For example, one of the simplest checks consists in determining the rate of increase of the output of an integrator for a constant input. On the other hand, one of the most informative but complicated dynamic checks is comparison of the computer solution with a solution that has been computed by numerical techniques of known accuracy. These numerically calculated solutions, or check solutions, provide the most satisfactory way yet discovered for securing complete assurance that the computer is operating correctly.

Generally, the dynamic checking methods require much more advanced preparation than the static checking methods. A complete check solution for a complicated set of differential equations may require several months to calculate by hand with a desk computing machine. Furthermore, such problems are difficult to solve on existing digital computing equipment because the storage and programing requirements become very great for problems of this class. The most complicated static checks, on the other hand, require at most a few hours of advance computation.

The majority of the dynamic checks used with the M.I.T. Flight Simulator fall into two classes, open-loop checks and closed-loop checks. This division arises from the fact that most of the problems solved on the computer are in the form of one or more major feedback loops plus numerous minor feedback channels. A complete check solution provides the most extensive closed-loop test. Here, the computer is operated in its normal manner with all feedback loops closed. However, it is also often convenient to break some of the feedback loops in one or more places and test the dynamic response either of an entire loop of the computer or merely of part of the loop. Tests with the loop thus broken are referred to as open-loop dynamic checks.

Closed-loop dynamic checks possess the advantage that they provide a better measure of the over-all operation of the computer than open-loop checks. Also more reliable numerical techniques are available for calculating closed-loop check solutions. On the other hand, with the feedback loops closed, errors generated by any components in the computer propagate through the entire computer. Thus it is very difficult to localize a malfunctioning component by means of closed-loop measurements alone. In open-loop dynamic checks the section of the computer in which the fault occurs is at once put in evidence, and it is possible to decrease systematically the number of components included in the dynamic check until the exact location of the fault is determined.

In making an open-loop check, a known driving function is inserted at the beginning of the section to be tested. The response of the section is measured and compared with the previously calculated correct response. The known driving function is selected with respect to its availability in the computer, its similarity to the type of signals occurring in the actual problem, and the ease with which the response of the open-loop section may be calculated. By a suitable choice of the driving function, a possibility exists for combining the advantages of both closed-loop and open-loop tests. Once a check solution has been calculated, then the correct signals throughout the entire computer are known. By making the open-loop driving function equal to the signal that would have existed at the same point in the closed-loop system, it is possible to achieve a number of advantages. In particular, such open-loop testing signals are very similar to the signals in the actual problem to be solved. Also, one numerically calculated check solution serves for both closed- and open-loop checks. The closed-loop signals are usually not sinusoids or other functions that can be generated by simple combinations of computing elements. As a consequence, the use of such driving functions requires a special input device for their generation, for example, a photoelectric input table or a digital-to-analogue data converter. Although such a device has not yet been tried with the M.I.T. Flight Simulator, it might prove to be valuable.

In summary, the general testing methods just described consist of static and dynamic checks. The static checks are inherently open-loop checks. The dynamic checks can be either open- or closed-loop checks. These checking methods are generally applicable to almost all analogue computers. By proper application of the checks, it is possible to localize quickly and eliminate faults due to human mistakes and component malfunctionings. After the faults have been removed, it is possible to maintain confidence that the computer is operating correctly by the periodic application of over-all checks such as the comparison of the computer results with check solutions.

In addition to the general methods, Trembath in his thesis[4] discusses a number of methods of error-checking specifically applicable to the simulation of three-dimensional flight paths. Because of their specific nature the methods were not discussed here, but their existence should be mentioned again because three-dimensional simulation is of great current interest.

7. SUGGESTIONS FOR FUTURE WORK

The direction which future error-analysis work should take will be discussed in this section. Several specific investigations will be outlined from the results of the preceding work.

7.1. Blind Alleys.

The results obtained for some of the work carried out last year show that certain investigations should be abandoned. The negative results are probably the most significant accomplishments because they show conclusively that no more effort should be put into certain areas. The studies of error generation in computer components, error propagation in problems, and error propagation in computers have demonstrated that from a practical standpoint it is useless to attempt to calculate the specific errors introduced into the solution of a problem. The calculation of specific errors requires so much time for problems of a practical size that it cannot be justified. Excessive time is required merely because so much data must be considered in calculating solution errors. Consequently, the use of large computing machines appears to offer the only means of shortening the calculations. In most cases, the use of such machines would defeat the purpose of the error-analysis program.

The studies of error propagation in problems and in computers have established that the error-propagation equations can be assumed to be linear. Thus certain general properties of linear equations hold for the error equations. In particular, because of superposition, it is possible to define a problem error sensitivity that is a function of the problem itself, and not a function of the specific way in which the problem is solved. In many practical problems, it is feasible to evaluate the sensitivity. Instead of trying to calculate problem errors, it may be sufficient to evaluate the sensitivity of the problem in order to estimate whether the problem can be solved satisfactorily on a given computer. Thus an investigation which should be pursued is the correlation of sensitivity to observed errors in problem solutions.

One way to establish the nature of the correlation would be to evaluate an empirical function that perhaps can be written in the form

Observed Solution Errors = Function (sensitivity, numbers of various kinds of computing elements used in computer).

The observed errors can be evaluated directly from a comparison of the computer solution and the check solution. The function should be kept as simple as possible. Thus, an attempt should be made to use some single, simple measure of the sensitivity together with a simple measure of how the problem is solved. such as the number of various kinds of computing elements used.

There are a number of error-propagation functions that make up the error sensitivity of a problem. The most unstable propagation function appears to be the most suitable single measure of the error sensitivity. Possibly a simple iteration process could be developed for finding the most unstable propagation function. If the linear error-propagation equations, Eqs. (9), are considered as a matrix equation, then the most unstable propagation function corresponds to the characteristic root of the matrix having the largest real part. Iteration methods for finding this characteristic root are particularly simple, and their use should be investigated as a means for shortening the work involved in calculating the sensitivity. A simple method for estimating sensitivity combined with an empirical function for predicting solution errors would provide a much needed rule of thumb for estimating solution errors.

7.2. Analyzing Operational Experience.

The greatest and most valuable progress in the work done so far on error analysis has been the development of practical operational techniques for locating errors in computers. Consequently, in future work greater emphasis should be placed on this area. The specific methods developed have been accumulated rather haphazardly by computer operators. Effective methods have been arrived at by trial and error. Ineffective processes have been gradually discarded. No particular effort has been made to evaluate directly the relative effectiveness of different methods. In order to increase the rate of development of good methods, a more orderly analysis and evaluation should be accomplished. A definite effort should be made to study the techniques used by the better operators and to reduce these techniques to specific written methods. The effectiveness of the specific written methods should then be evaluated through their application by less experienced operators.

A particular technique that appears to be promising and that should be developed further is the use of closed-loop test signals for open-loop tests. The advantages of such a procedure were outlined in Sec. 6.3. The process requires the use of some sort of input-data device that can convert an arbitrary tabulation or plot of the closed-loop signals into electrical computer signals. Arbitrary-function generators can be used for this purpose. A bank of these generators will be installed shortly in the M.I.T. Flight Simulator and should provide an opportunity to obtain some practical experience in using the open-loop tests.

7.3. General Error-Probability Function.

The principal reason for the failure to attain the originally desired goal of the studies of error generation and propagation was the attempt to obtain too detailed information concerning the errors. At first, it was desired to estimate the specific errors as time functions. As might have been expected, obtaining such detailed information requires too much calculation to be of practical use. A more practicable objective would be the determination of the average or statistical properties of the errors. Most types of errors are analyzed on a statistical basis. Indeed, considerable statistical theory first was developed as a means of treating errors. Also, the analysis of error generation in computer components showed that while it is not feasible to estimate the actual time-function component errors, some of the statistical properties of the errors can be measured and calculated relatively easily. As a consequence of these developments, an attempt should be made to set up and evaluate a general error-probability function.

Such a function would give the probability of solution errors as a function of the problem being solved and of the method used for its solution. The function could include effects of the probability of component failure and the probability of operator mistakes as well as the effects of random errors generated by the computer components.

The effects of random component errors can be included easily in a general probability function. Nevertheless, it is not easy to see how the effects of systematic component errors can be included. Some sort of approximation will have to be made to randomize the total effect of a number of systematic errors. However, if the number of systematic errors is large, as it will be for a problem using many computer components, then the assumption of random effects may be reasonable.

In evaluating the general error-probability function, the fact that errors propagate linearly should be of considerable value because a large amount of statistical theory that is applicable to this case has been developed recently. The linearity of the propagation greatly reduces the amount of data that must be known about generated errors in order to obtain the statistical characteristics of the output errors.

APPENDIX A. BIBLIOGRAPHY

1. M. V. Mathews, Study of Error Generation in Analogue Computer Components. Thesis (S.M.), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1952.

2. H. Mori, Analysis of Numerical Solutions for the M.I.T. Flight Simulator. Thesis (S.M.), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1952.

3. G. Rabow, Evaluation of Errors Occurring in the M.I.T. Flight Simulator. Thesis (S.M.), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1952.

4. N. W. Trembath, Simulation of Three-Dimensional Flight Paths. Thesis (S.M.), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1953.

5. F. J. Murray and P. Brock, Planning and Error Analysis for the Numerical Solution of a Test System of Differential Equations on the IBM Sequence Calculator, (Project Cyclone Report to Special Devices Center of Office of Naval Research, Contracts Nos. N6ori-128 and N6140s-2890B). New York, N. Y.: Reeves Instrument Corporation, Problem concluded March, 1949.

6. T. F. Jones, Jr., The Propagation of Errors in Analogue Computers. Thesis (Sc.D), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1952.

7. R. C. Booton, Jr., Nonlinear Control Systems with Statistical Inputs, Dynamic Analysis and Control Laboratory Report No. 61. Cambridge, Mass.: Mass. Inst. of Tech., March 1, 1952.

8. R. C. Booton, Jr., W. W. Seifert, and M. V. Mathews, Nonlinear Servomechanism with Random Inputs, Dynamic Analysis and Control Laboratory Report No. 70. Cambridge, Mass.: Mass. Inst. of Tech., forthcoming 1953.

9. E. C. Johnson, Jr., Sinusoidal Analysis of Feedback-Control Systems Containing Nonlinear Elements. Thesis (Sc.D.), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1951.

10. R. J. Kochenburger, Analysis and Synthesis of Contractor Servomechanisms. Thesis (Sc.D), Dept. of Elec. Engr., Mass. Inst. of Tech., Cambridge, Mass., 1949.